CMPE-445 INTERNET ARCHITECTURE AND PROTOCOLS

Laboratory Work

PART 1: Using the Ping program to investigate connections between hosts in the Internet

I. INTRODUCTION

The Ping program is one of the utilities in UNIX and Window. The name “ping” is taken from the sonar operation to locate objects. The Ping program tests whether another host is reachable. If the host is reachable, the program can provide data on time delays and lost packets in the communication with the remote host.

As in all client-server systems, the Ping program implements the tasks of a client and a server. Typically, the server part of the program runs all the time on each host (if there is no firewall at the host). However, the client part must be started explicitly by a user.

Working as a client, the Ping program sends echo requests **once per second** to a selected remote host, which plays the role of a server. For each received echo request, the server returns the received message as an echo reply (without changing it).

After receiving an echo reply, the client calculates time delay for the corresponding request. The client also calculates lost packets, i.e. the requests for which the client could not receive an echo reply from the server. Packets, sent by the client, may not only be lost, but be duplicated or arrive out of order because the Ping program uses the UDP protocol that is not reliable.

The goal of the laboratory work is to use the Ping program for performance investigation of connection between different pairs of hosts in the Internet. During the laboratory work, the student is required to run the Ping program for different remote hosts and fix, in his/her notebook, results of execution of the program. Using the collected data, the student should take his/her conclusions relating to data exchange between hosts in the Internet. **The student should get an idea on the typical round trip times between hosts in the Internet.**

Before starting experiments, a student should read in UNIX, using the shell command **man**, the description of the Ping program, its options and parameter(s). Note that this description may be different for different UNIX systems.

II. EXPERIMENTS

1. Login to a UNIX (or Linux) system.

2. Start the Ping program **four times** with the default packet size (56 + 8 bytes) and the number of packets not less than 100. The corresponding command line is as follows:

```
% ping  -c  count  -s   packetsize  destination_host_name_or_IP_address
```

Use one remote host from **each** of the following four geographical areas:

(a) a host in this university (cmpe.emu.edu.tr, ee.emu.edu.tr, ie.emu.edu.tr, etc.);
(b) a host in Turkey;
(c) a host in Europe;
(d) a host somewhere else (USA, Russia, Canada, Japan, Australia).

Choose appropriate host names or IP addresses from the list provided in Appendix.

3. Fix the results of the four experiments in a table of the following format:

Number of packets sent = Packet size = bytes

Domain name or IP address of a remote host	Geographical site of the host (country)	Number of sent packets	Packets lost		Round trip time of a packet, ms		
			Number	%	Min	Ave	Max

3. Repeat experiments 2 and 3 **three times** with the increased packet size (500, 1000, 2000 and 4000 bytes). Use the **same addresses** of remote hosts as in the experiments 2 and 3. Fix the results in the tables of the above given format.

Number of packets sent = Packet size = bytes

Domain name or IP address of a remote host	Geographical site of the host (country)	Number of sent packets	Packets lost		Round trip time of a packet, ms		
			Number	%	Min	Ave	Max

Number of packets sent = Packet size = bytes

Domain name or IP address of a remote host	Geographical site of the host (country)	Number of sent packets	Packets lost		Round trip time of a packet, ms		
			Number	%	Min	Ave	Max

Number of packets sent = Packet size = bytes

Domain name or IP address of a remote host	Geographical site of the host (country)	Number of sent packets	Packets lost		Round trip time of a packet, ms		
			Number	%	Min	Ave	Max

Number of packets sent = Packet size = bytes

Domain name or IP address of a remote host	Geographical site of the host (country)	Number of sent packets	Packets lost		Round trip time of a packet, ms		
			Number	%	Min	Ave	Max

4. Compare all four tables of the resulting statistics. Make your conclusions and fix them in your notebook.

III. REQUIREMENTS

1. A student must clearly understand the goal of the Ping program, its options and parameters and how to use the program.

2. In the report, a student should explain why the program can be useful (and in what circumstances).

3. The report must contain
   (a) a short description of the Ping program;
   (b) a command line for starting the program, with the explanation of options and parameters;
   (c) **four tables** of results of running the program;
   (d) a summary of the results.

4. A student must be able to answer the following questions.

IV. QUESTIONS

1. What is the purpose of the Ping program?
2. In which situations is the program most useful?
3. What mechanism of interprocess communication is used by the program?
4. What protocol is used by the program?
5. What can happen with data packets sent by the program?
6. What is the size of the header in data packets sent by the program?
7. What is a default period of sending echo requests?
8. Write and explain a command line to start the Ping program, with collecting the statistics?
9. In what mode does the Ping program operate on a remote computer that is a target of the Ping program?
10. What forms of addressing of remote hosts can be used by the program?
11. What is the typical average round trip time of packets sent by the program
    (a) to a remote host in this university?
    (b) to a host in Turkey?
    (c) to a host somewhere else?

The source of information:

UNIX, use the shell command **man** to get the description of the Ping program.

PART 2: Using the Traceroute program for tracing the routes of datagrams in the Internet

I. INTRODUCTION

The Traceroute program is a debugging tool that allows you to explore the TCP/IP protocols. Although there are no guarantees that two consecutive IP datagrams from the same source to the same destination follow the same route, most of the time they do. Traceroute lets you see the route that IP datagrams follow from one host to another.

Traceroute uses UDP and ICMP protocols (on ICMP, see the textbook of Tanenbaum, A., pp. 419 - 420). It sends IP datagrams, with UDP field in the IP header. When a datagram moves in the Internet from a source host to a destination host, it passes a number of intermediate routers. Each router decrements the TTL field in datagrams by one, so the TTL field can be considered as a hop counter. "Hop" means an intermediate node in the Internet on the way from a source to destination.

When a router gets an IP datagram whose TTL is 1, it decrements TTL (so it becomes zero) AND does not forward this datagram but discards it AND sends back to the originating host an ICMP "time exceeded" error message. Therefore, in this case the response from the router is sent back as an IP datagram, with ICMP (not UDP) protocol in the IP header. Thus the program sends IP request datagrams with UDP protocol, but receives IP diagnostic datagrams with ICMP protocol.

Traceroute operates in the following way. Having IP address of destination host, it sends an IP datagram with TTL = 1 to the destination host. The first router on the way decrements the TTL (so it becomes zero), discards the datagram, and sends back to the source host an ICMP “time exceeded” error message, with IP address of this router. After receiving this datagram, Traceroute identifies the first router and prints its domain name and IP address. Then Traceroute sends a datagram to the same destination host with TTL = 2, and as a result it will know the second router. This continues until the datagram reaches the destination host, so that Traceroute will determine all routers on the way to that destination.

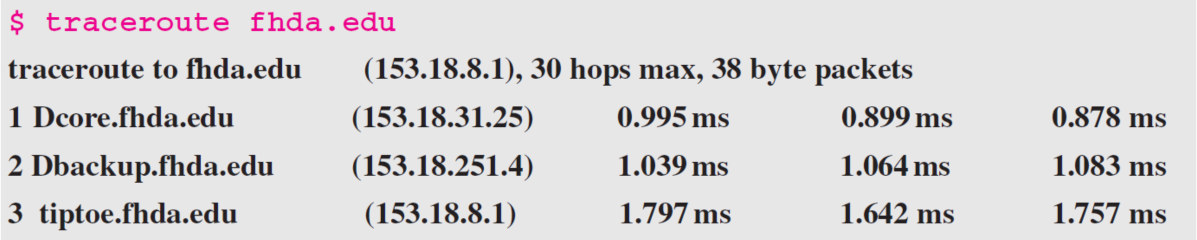
The purpose of the laboratory work is to use the Traceroute program for tracing routes (paths) of datagrams between your source host (helium, for example) and different destination hosts in the Internet. During the laboratory work, a student is required to run the Traceroute program for different remote hosts and fix, in his/her notebook, results of execution of the program. Using the collected data, a student should take his/her conclusions relating to moving datagrams between hosts in the Internet. Note that some or even all datagrams sent by Traceroute can be lost for some value of TTL.

Before starting experiments, a student should read in UNIX, using the shell command **man**, the description of the Traceroute program, its options and parameter(s). Note that this description can be different for different UNIX systems.

II. EXPERIMENTS

1. Login to a UNIX or Linux system and start the Traceroute program with the name of a destination host in this University (for example, cmpe.emu.edu.tr, ee.emu.edu.tr, ie.emu.edu.tr, etc) as a parameter. You will have something like the following:

%traceroute fhda.edu (an IP address may be used instead of the domain name as well)



```
$ traceroute fhda.edu
traceroute to fhda.edu      (153.18.8.1), 30 hops max, 38 byte packets
1 Dcore.fhda.edu            (153.18.31.25)    0.995 ms    0.899 ms    0.878 ms
2 Dbackup.fhda.edu          (153.18.251.4)    1.039 ms    1.064 ms    1.083 ms
3  tiptoe.fhda.edu           (153.18.8.1)     1.797 ms    1.642 ms    1.757 ms
```

The first unnumbered line of output shows the complete domain name and IP address of the desired destination host and indicates the program won't increase the TTL beyond 30 (not more than 30 routers on the way to the destination). The datagram size of 38 bytes allows for the 20-byte IP header, the 8-byte UDP header, and 10 bytes of used data (the 10 bytes of used data contain a sequence number that is incremented each time a datagram is sent, a copy of the outgoing TTL, and the time at which the datagram was sent).

The next two lines in the output begin with the TTL value, followed by the name of the host or router, and its IP address. For each TTL value *three datagrams are sent* by the Traceroute. Correspondingly, for each returned ICMP message the round-trip time is calculated and printed. If no response is received within 5 seconds for a datagram, an asterisk is printed instead and the next datagram is sent.

The round-trip times (0.995 ms, 0.899 ms, and 0.878 ms for the router router1.emu.edu.tr in our example) are calculated on the sending host. Typically, the round-trip time increases from one router to another, but it is NOT the general case.

If, starting with some TTL value, the Traceroute prints only asterisks, then it means that some router (or routers) on the way to the destination host discards all IP datagrams. The reason can be the overload of the router at this time.

2. Start the Traceroute **four more times**, using one address of a destination host from **each** of the following geographical areas:

(a) another host in this university (which is not selected in step 1);
(b) a host in Turkey;
(c) a host in Europe;
(d) a host somewhere else (USA, Canada, Japan, Australia).

Choose appropriate destination host names from the list given in Appendix.

3. Fix the results of the four experiments in a table of the following format:

Source host is:…………………………………

Domain name or IP address of a destination host	**Geographical site of the host (country)**	**Numb. of hops to destin. host**	**Name and IP address of each router (on the way to destination)**	**Round trip times for the router, ms** **1st 2nd 3rd**

III. REQUIREMENTS

1. A student must clearly understand the goal of the Traceroute program, its parameters and how to use the program.

2. In the report, a student should explain why the program can be useful (and in what circumstances).

3. The report must contain
   (a) a short description of the Traceroute program;
   (b) a command line for starting the program, with the explanation of parameters;
   (c) a table of results of running the program;
   (d) a short summary of the results.

   (a) and (b) must be prepared **before** the laboratory work (at home).

4. A student must be able to answer the following questions.

IV. QUESTIONS

1. What is the purpose of the Traceroute program?
2. In which situations is the program most useful?
3. How does the program work?
4. What transport protocol (TCP? UDP?) is used by the program?
5. What can happen with IP packets (datagrams) sent by the program?
6. What is the size of packets sent by the program?
7. How long does the program waits for a response after sending an IP packet? (optional question)
8. What does the program print if there is no response?
9. In what mode does the Ping program operate on a remote computer that is a target of the Ping program?
10. What forms of addressing of destination hosts can be used to start the program?
11. What is a hop?
12. Suppose that there are 10 routers on the way to a destination host. How many IP packets will be sent by the program?
13. What is TTL?
14. What is the typical maximal TTL value printed by the program for a destination host
    (a) in this university?
    (b) in Turkey?
    (c) somewhere else?

The source of information:

1. UNIX, use the shell command **man** to get the description of the Traceroute program.
2. Tanenbaum, A.S. Computer Networks, 3rd ed., pp. 419 - 420.
3. Stevens, W.R., TCP/IP Illustrated, vol. 1, pp. 97 - 110.

APPENDIX: A list of addresses of remote hosts in the Internet

For each of the hosts, its address and site (country or university or firm) are given.

metu.edu.tr	METU, Ankara
itu.edu.tr	ITU, İstanbul
193.140.83.1	ULAK net, Turkey
cs_dept_server_1.uhh.hawaii.edu	Univ. of Hawaii
agora.leeds.ac.uk	Univ. of Leeds (UK)
csri.toronto.edu	Univ. of Toronto (Ca)
infolib.murdoch.edu.au	Murdoch Univ. (Austra)
gopher.novell.com	Novell
ora.com	O'Reilly & Assoc.
www1.cern.ch	CERN
coral.cs.jcu.edu.au	Australia
www.cs.bilkent.edu.tr	Bilkent Univ.
www.inesc.pt	Portugal
www.csc.fi	Finland
www.crl.go.jp	Japan
debra.dgbt.doc.ca	Ottawa (Canada)
www.crs4.it	Sardinia
services.csp.it	Italy
boole.stanford.edu	Stanford Univ.
iamftp.unibe.ch	Bern Univ.
ghost.dsi.unimi.it	Milan Univ.
owl.pmms.cam.ac.uk	Cambridge Univ.
camel.cecm.sfu.ca	Canada
gopher.can.nl	Netherland
snekkar.ens.fr	France
newton.newton.cam.ac.uk	UK
gopher.qmw.ac.uk	London
cfata9.harvard.edu	Harvard Univ.
henri.ma.utexas.edu	Univ. of Texas
cs-ftp.bu.edu	Boston Univ.
cs.nyu.edu	New York Univ.
cs.rice.edu	Rice Univ.
www.ecrc.de	Germany
cs.nyu.edu	New York University
google.com	
amazon.com	
facebook.com	
www.gmd.de	Germany
www.hut.fi	Helsinki
www.glocom.ac.jp	Japan
www.iss.nus.sg	Singapore
ai.iit.nrc.ca	Canada
zenon.inria.fr	France
www.wi.leidenuniv.nl	Leiden Univ.
www.dna.lth.se	Sweden
www.cs.monash.edu.au	Australia
ftp.novell.de	Germany
www.nada.kth.se	Stockholm
web.cs.ualberta.ca	Canada
www.cs.bham.ac.uk	UK
www.dcc.uchile.cl	Chile
www.cs.man.ac.uk	UK
ftp.cs.umanitoba.ca	Canada
www.iro.umontreal.ca	Canada
www.cs.uit.no	Norway
www.nns.ru	Russia
www.izvestia.ru	Russia
www.center.ru	Russia
www.cdi.org	USA